NULL GOEDESICS IN KERR AND SCHWARZSCHILD SPACE-TIME

Eunice Monyenye Omwoyo
Supervised by Prof. Marc Casals

OVERVIEW

- Brief introduction
- Explanation of Kerr metric and the associated null geodesics.
- Formulation of the code
- Visualize sample plots using the code
- Conclusion

INTRODUCTION

- In General Relativity, a geodesic generalizes the notion of a straight line to curved space-time.
- A massless test particle in motion on a certain space-time with no forces acting on it follows a trajectory called a null geodesic.
- Null geodesics are crucial in understanding the nature of black holes.
- For instance, recently, the Event Horizon Telescope showed the image of a shadow of the supermassive black hole M87*. The shadow of a black hole is formed by null geodesics.

KERR METRIC

- The Kerr metric,in Boyer- Lindquist coordinates(t,r,θ,ϕ), has the form:

$$ds^2 = -\left(1 - \frac{2\,\mathrm{Mr}}{\Sigma}\right) dt - \frac{4\,M\,a\,r\,\sin^2\theta}{\Sigma}\, d\phi dt + \frac{\Sigma}{\Delta}\, dr^2 + \Sigma d\theta^2 + \left(r^2 + a^2 + \frac{2\,\mathrm{Mra}^2\,\mathrm{Sin}^2\,\theta}{\Sigma}\right) \sin^2\theta\, d\phi^2;$$

$$\Sigma = r^2 + a^2 \cos^2\theta;\ \Delta = r^2 - 2\,\mathrm{Mr} + a^2$$

- The metric admits two killing vectors ∂_t and ∂_ϕ hence it is stationary and axially symmetric.
- In the limit a→0, the Kerr metric reduces to a Schwarzschild metric.
- Motion in Kerr space-time is governed by constants of motion E(related to the geometry being stationary), L(related to the axial symmetry), Q(a hidden symmetry. It arises from separation of variables in Hamilton-Jacobi equation).
- The constants of motion in used in this work have been rescaled as

  $\lambda = \frac{l}{E}, \eta = \frac{Q}{E^2}$
- Null geodesic equations in Kerr space-time are described by the equations; (Null geodesics of the Kerr exterior
  Samuel E. Gralla and Alexandru Lupsasca
  Phys. Rev. D 101, 044032)

$$\frac{\Sigma}{E}\, p^r = \pm_r \sqrt{R(r)}$$

$$\frac{\Sigma}{E}\, p^\theta = \pm_\theta \sqrt{\Theta(\theta)}$$

$$\frac{\Sigma}{E}\, p^\phi = \frac{a}{\Delta}\left(r^2 + a^2 - a\,\lambda\right) + \frac{\lambda}{\sin^2\theta} - a$$

$$\frac{\Sigma}{E}\, p^t = \frac{\left(r^2 + a^2\right)}{\Delta}\left(r^2 + a^2 - a\,\lambda\right) + a\left(\lambda - a\sin^2\theta\right)$$

$$p^\mu = \frac{d\,x^\mu}{d\sigma};\ \sigma \text{ is the affine parameter}$$

$$R(r) = \left(r^2 + a^2 - a\lambda\right)^2 - \Delta(r)\left(\eta + (\lambda - a)^2\right)$$

$$\Theta(\theta) = \eta + a^2\cos^2\theta - \lambda^2\cot^2\theta$$

- New parametrization, “Mino time” τ, defined as

$$\frac{dx^\mu}{d\tau} = \frac{\Sigma}{E}\, p^\mu$$

FORMULATION OF THE CODE

We have formulated the code using (Null geodesics of the Kerr exterior
Samuel E. Gralla and Alexandru Lupsasca
Phys. Rev. D 101, 044032).

- All that the user needs is a given set of initial positions x^μ , initial momentum p^μ together with the commands that we shall define to evaluate and analyze various properties of these Null geodesics.

(a, ts, rs, θs, ϕs, p^rs, p^θs, p^ϕs).

- We first define the code such that ;

$p_\mu \, p^\mu = 0$

- We then calculate the constants of motion, λ and η

CONSTANTS OF MOTION CODE

Examples for constants of motion

- To evaluate constants of motion, the user needs the command: **ConstantsOfMotion**

In[4]:=
```
ConstantsOfMotion [0.9 , 0, 13 , π / 2, 0, -476 , 2, 1]
```
Out[4]= $\{\lambda \to 0.193859 , \eta \to 0.505469 \}$

In[5]:=
```
ConstantsOfMotion [0.9 , 0, 18 , π / 2, 0, -476 , 0, 1](*equatorial orbit Kerr *)
```
Out[5]= $\{\lambda \to 0.570609 , \eta \to 0.\}$

In[6]:=
```
ConstantsOfMotion [0, 0, 13 , π / 2, 0, -50 , 0, 1](*equatorial orbit schwarzschild *)
```
Out[6]= $\left\{\lambda \to \frac{169}{\sqrt{2643}}, \eta \to 0\right\}$

In[7]:=
```
ConstantsOfMotion [0, 0, 13 , π, 0, -476 , 0, 1](*polar orbit schwarzchild *)
```
Out[7]= $\{\lambda \to 0, \eta \to 0\}$

In[8]:=
```
ConstantsOfMotion [0.5 , 0, 13 , π, 0, -476 , 0, 1](*a polar orbit kerr *)
```
Out[8]= $\{\lambda \to 0., \eta \to -0.25\}$

In[9]:=
```
ConstantsOfMotion [0, 0, 3, π / 2, 0, 0, 0, 1](*schwarzschild photon sphere *)
```
Out[9]= $\left\{\lambda \to 3 \sqrt{3}, \eta \to 0\right\}$

RADIAL MOTION

- The radial potential is given by;

$$R(r) = (r^2 + a^2 - a\lambda)^2 - \Delta(r)\,(\eta + (\lambda - a)^2)$$

- This potential has 4 roots(r1,r2,r3,r4) which can be complex or real.
- The roots will be the turning points for the null geodesics.
- The nature of the roots will determine the various cases of radial motion.

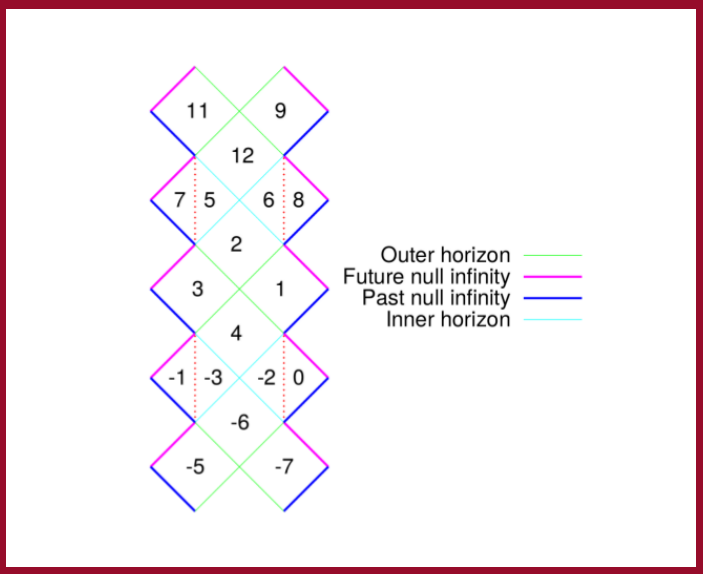
ROOTS OF THE RADIAL POTENTIAL CODE

EXAMPLES OF RADIAL POTENTIAL ROOTS

- The radial potential roots will be calculated using the command, **KerrNullGeoRadialRoots**

case one: r1<r2<rs<r3<r4

In[11]:=



In[12]:= `KerrNullGeoRadialRoots [0.3, 0, 2.4, 2.7, 0, 4, 7.1, 20]`

Out[12]= $\{r_1 \to -5.76165,\ r_2 \to 0.0430491,\ r_3 \to 2.73615,\ r_4 \to 2.98245\}$

case two: r1<r2<r3<r4<rs

In[13]:= `KerrNullGeoRadialRoots [0.8, 0, 14, π/2.3, 0, 200, 0, -10]`

Out[13]= $\{r_1 \to -9.24444,\ r_2 \to 0.0109996,\ r_3 \to 2.70194,\ r_4 \to 6.53149\}$

In[14]:= `KerrNullGeoRadialRoots [0.5, 0, 17, π/2, 0, -120, 0, 9]`

Out[14]= $\{r_1 \to -14.679,\ r_2 \to -1.77636 \times 10^{-15},\ r_3 \to 1.89616,\ r_4 \to 12.7828\}$

case three: r1<r2<rs, r3=r4*

In[15]:=
```
KerrNullGeoRadialRoots [0.9, 0, 2, 3, 0, -5, 10, 8]
```

Out[15]= $\{r_1 \to -5.38943,\ r_2 \to 0.506942,\ r_3 \to 2.44124 - 0.549201\, i,\ r_4 \to 2.44124 + 0.549201\, i\}$

case four: r1=r2*, r3=r4*

In[16]:=
```
KerrNullGeoRadialRoots [0.2, 0, 10, 0.3, 0, -10, 0, 0]
```

Out[16]= $\{r_1 \to -0.08033 - 0.103099\, i,\ r_2 \to -0.08033 + 0.103099\, i,$
$r_3 \to 0.08033 - 0.280294\, i,\ r_4 \to 0.08033 + 0.280294\, i\}$

double roots

In[17]:=
```
KerrNullGeoRadialRoots [0.5, 0, 2.8832177419263525 , 0, 0, 0, 20, 10]
```

Out[17]= $\{r_1 \to -5.89884,\ r_2 \to 0.1324,\ r_3 \to 2.88322,\ r_4 \to 2.88322\}$

In[18]:=
```
KerrNullGeoRadialRoots [0, 0, 3, 0, 0, 0, 20, 10]
```

Out[18]= $\{r_1 \to -6,\ r_2 \to 0,\ r_3 \to 3,\ r_4 \to 3\}$

Radial motion code

EXAMPLES OF RADIAL MOTION

- To evaluate radial motion, the user needs the command, **RadialMotion**

In[71]:=
```
RadialMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]["RadialRoots "]
```
Out[71]= $\{r_1 \to -5.99652 , r_2 \to 0.0429647 , r_3 \to 2.45168 , r_4 \to 3.50187 \}$

In[320]:=
```
RadialMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]
```
Out[320]=
```
case1Function [0.3 ,0,2.4 ,2.7 ,0,2.1 ,7.1 ,20 ,<<>>]
```

In[72]:=
```
RadialMotion [0.3 , 0, 2.4 , 2.7 , 0, 3, 7.1 , 20]["RadialRoots "]
```
Out[72]= $\{r_1 \to -5.90006 , r_2 \to 0.0429986 , r_3 \to 2.52155 , r_4 \to 3.33551 \}$

In[73]:=
```
RadialMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]["ConstantsofMotion  "]
```
Out[73]= $\{\lambda \to 1.62165 , \eta \to 24.5772 \}$

In[327]:=
```
RadialMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20][τ](*to visualize  the equation  used*)
```
Out[327]=
$$\frac{0.362974 + 14.4439\ \text{JacobiSN}[2.70285\ (0.723386 + \tau), 0.782948]^2}{8.4482 - 2.40872\ \text{JacobiSN}[2.70285\ (0.723386 + \tau), 0.782948]^2}$$

In[75]:=
```
RadialMotion [0.8 , 0, 14 , π / 2.3 , 0, 200 , 0, -10][τ]
```
Out[75]=
$$\frac{78.0277 - 42.6257\ \text{JacobiSN}[4.41294\ (0.188295 + \tau), 0.544983]^2}{11.9464 - 15.7759\ \text{JacobiSN}[4.41294\ (0.188295 + \tau), 0.544983]^2}$$

In[76]:=
```
RadialMotion [0, 0, 14 , π / 2, 0, 200 , 0, 10]
```
Out[76]=
```
case2Function [0,0,14 ,Pi
--
2,0,200 ,0,10 ,<<>>]
```

In[77]:=
```
RadialMotion [0.9 , 0, 2, 3, 0, -5, 10 , 8]
```
Out[77]=
```
case3Function [0.9 ,0,2,3,0,-5,10 ,8,<<>>]
```

In[78]:=
```
RadialMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]
```
Out[78]=
```
case4Function [0.2 ,0,10 ,0.3 ,0,-10 ,0,0,<<>>]
```

POLAR MOTION

The angular potential is also evaluated to arrive at four roots ($\theta 1$,$\theta 2$,$\theta 3$,$\theta 4$).

The nature of these roots lead to two cases of polar motion.

ANGULAR POTENTIAL ROOTS

Example for angular potential roots

The command **KerrNullGeoAngularRoots** is used to calculate the roots of the radial potential

ordinary motion: two real roots $\theta_1 \leq \pi/2 \leq \theta_4$ and is characterized by $\eta \geq 0$

In[80]:=
```
KerrNullGeoAngularRoots [0.3, 0, 2.4, 2.7, 0, 2.1, 7.1, 20]
```
Out[80]= $\{\theta_1 \to 0.315649, \theta_2 \to 1.5708 - 3.54952\, i, \theta_3 \to 1.5708 + 3.54952\, i, \theta_4 \to 2.82594\}$

In[81]:=
```
ConstantsOfMotion [0.3, 0, 2.4, 2.7, 0, 2.1, 7.1, 20]
```
Out[81]= $\{\lambda \to 1.62165, \eta \to 24.5772\}$

In[82]:=
```
KerrNullGeoAngularRoots [0.8, 0, 14, π/2.3, 0, 200, 0, -10]
```
Out[82]= $\{\theta_1 \to 1.36591, \theta_2 \to 1.5708 - 3.02646\, i, \theta_3 \to 1.5708 + 3.02646\, i, \theta_4 \to 1.77568\}$

In[83]:=
```
KerrNullGeoAngularRoots [0.8, 0, 14, π/2, 0, 200, 0, -10]
(*confined within the equatorial plane*)
```
Out[83]= $\{\theta_1 \to 1.5708, \theta_2 \to 1.5708 - 3.04165\, i, \theta_3 \to 1.5708 + 3.04165\, i, \theta_4 \to 1.5708\}$

In[84]:=
```
ConstantsOfMotion [0.8, 0, 14, π/2, 0, 200, 0, -10]
```
Out[84]= $\{\lambda \to -8.39503, \eta \to 0.\}$

vortical motion: $\theta_1 \leq \theta_2 \leq \pi/2 \leq \theta_3 \leq \theta_4$ and is characterized by $\eta \leq 0$

In[85]:=
```
KerrNullGeoAngularRoots [0.2, 0, 10, 0.3, 0, -10, 0, 0]
```
Out[85]= $\{\theta_1 \to 0.0739136, \theta_2 \to 0.3, \theta_3 \to 2.84159, \theta_4 \to 3.06768\}$

In[86]:=
```
ConstantsOfMotion [0.2, 0, 10, 0.3, 0, -10, 0, 0]
```
Out[86]= $\{\lambda \to -0.00436462, \eta \to -0.0363076\}$

POLAR MOTION CODE

EXAMPLES OF POLAR MOTION

To analyze polar motion, the command **PolarMotion** is used,

In[129]:=
```
PolarMotion [0.3, 0, 2.4, 4, 0, 2.1, 7.1, -10]["AngularRoots "]
```
Out[129]= $\{\theta_1 \to 0.664203, \theta_2 \to 1.5708 - 3.78837\, i, \theta_3 \to 1.5708 + 3.78837\, i, \theta_4 \to 2.47739\}$

In[321]:=
```
PolarMotion [0.3, 0, 2.4, 4, 0, 2.1, 7.1, -10]
```
Out[321]= ordinaryFunction [0.3,0,2.4,4,0,2.1,7.1,-10,<<>>]

In[326]:=
```
PolarMotion [0.3, 0, 2.4, 4, 0, 2.1, 7.1, -10][τ](*to visualize the equation used*)
```
Out[326]= ArcCos [-0.787408 JacobiSN [6.62427 (0.147814 + τ), -0.00127165]]

In[130]:=
```
PolarMotion [0.3 , 0, 2.4 , 4, 0, 2.1 , 7.1 , -10]["ConstantsofMotion  "]
```
Out[130]=
```
{λ → -4.0876 , η → 27.2067 }
```

In[131]:=
```
PolarMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]["ConstantsofMotion  "]
```
Out[131]=
```
{λ → -0.00436462 , η → -0.0363076 }
```

In[132]:=
```
PolarMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]
```
Out[132]=
```
vorticalFunction [0.2 ,0,10 ,0.3 ,0,-10 ,0,0,<<>>]
```

In[133]:=
```
PolarMotion [0.8 , 0, 14 , π / 2, 0, 200 , 0, -10]
```
Out[133]=
```
ordinaryEquatorialFunction  [0.8 ,0,14 ,Pi
--
2,0,200 ,0,-10 ,<<>>]
```

AZIMUTHAL MOTION CODE

EXAMPLES OF AZIMUTHAL MOTION

- In the analysis of Azimuthal motion, the command **AzimuthalMotion** is used. The code will return various cases of azimuthal motion, for instance we denote η>0 with ηp; η<0 with ηm and we use 1,2,3,4to denote the four cases of radial potential roots.

```
In[216]:= AzimuthalMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]["ConstantsofMotion "]
Out[216]= {λ → 1.62165 , η → 24.5772 }

In[322]:= AzimuthalMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]
Out[322]= ϕηp1Function [0.3 ,0,2.4 ,2.7 ,0,2.1 ,7.1 ,20 ,<<>>]

In[325]:= AzimuthalMotion [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20][τ](*to visualize the equation used*)
Out[325]= 1.62165 (-0.47067 + 0.191747 EllipticPi [0.903631 ,
        JacobiAmplitude [5.2152 (0.240821 + τ), -0.00299015 ], -0.00299015 ]) +
  0.314485 (1.71069 (0.0149885 - 0.125779 (0.723386 + τ) - 0.147072
          EllipticPi [1.1862 , JacobiAmplitude [2.70285 (0.723386 + τ), 0.782948 ], 0.782948 ]) +
     0.197186 (-0.0444135 - 0.165492 (0.723386 + τ) - 119.438
          EllipticPi [556.454 , JacobiAmplitude [2.70285 (0.723386 + τ), 0.782948 ], 0.782948 ]))

In[217]:= KerrNullGeoCOM [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]
Out[217]= {λ → 1.62165 , η → 24.5772 }

In[218]:= KerrNullGeoRadialRoots [0.3 , 0, 2.4 , 2.7 , 0, 2.1 , 7.1 , 20]
Out[218]= {r1 → -5.99652 , r2 → 0.0429647 , r3 → 2.45168 , r4 → 3.50187 }

In[219]:= AzimuthalMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]["ConstantsofMotion "]
Out[219]= {λ → -0.00436462 , η → -0.0363076 }

In[220]:= AzimuthalMotion [0.9 , 0, 2, 3, 0, -5, 10 , 8]
Out[220]= ϕηp3Function [0.9 ,0,2,3,0,-5,10 ,8 ,<<>>]

In[221]:= AzimuthalMotion [0.8 , 0, 14 , π / 2.3 , 0, 200 , 0, -10]
Out[221]= ϕηp2Function [0.8 ,0,14 ,1.36591 ,0,200 ,0,-10 ,<<>>]

In[331]:= AzimuthalMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]
Out[331]= ϕηm4Function [0.2 ,0,10 ,0.3 ,0,-10 ,0,0 ,<<>>]

In[330]:= AzimuthalMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]["ConstantsofMotion "]
Out[330]= {λ → -0.00436462 , η → -0.0363076 }

In[328]:= AzimuthalMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]["RadialRoots "]
Out[328]= {r1 → -0.08033 - 0.103099 i, r2 → -0.08033 + 0.103099 i,
  r3 → 0.08033 - 0.280294 i, r4 → 0.08033 + 0.280294 i}
```

TEMPORAL MOTION

EXAMPLES FOR TEMPORAL MOTION

- In the analysis of Temporal Motion, the command **TemporalMotion** is used. The code will also return various cases of temporal motion, for instance we denote η>0 with ηp; η<0 with ηm and we use 1,2,3,4to denote the four cases of radial potential roots.

In[336]:=
```
TemporalMotion [0.3 , 0, 2.4 , 2.7 , 0, -2.1 , 7.1 , 20]
```
Out[336]=
```
tηp1Function [0.3 ,0,2.4 ,2.7 ,0,-2.1 ,7.1 ,20 ,<<>>]
```

In[332]:=
```
TemporalMotion [0.3 , 0, 2.4 , 2.7 , 0, -2.1 , 7.1 , 20]["RadialRoots "]
```
Out[332]=
```
{r1 → -5.99652 , r2 → 0.0429647 , r3 → 2.45168 , r4 → 3.50187 }
```

In[334]:=
```
TemporalMotion [0.3 , 0, 2.4 , 2.7 , 0, -2.1 , 7.1 , 20][τ](*to visualize  the equation  used*)
```
Out[334]=
```
1.56816 - 10.4468 (0.723386 - τ) + 4 τ +
 5.40569 EllipticE [JacobiAmplitude [2.70285 (0.723386 - τ), 0.782948 ], 0.782948 ] +
 0.09 (-0.083332 +
    57.9466 (EllipticE [JacobiAmplitude [5.2152 (0.240821 + τ), -0.00299015 ], -0.00299015 ] -
       EllipticF [JacobiAmplitude [5.2152 (0.240821 + τ), -0.00299015 ], -0.00299015 ])) +
 2 (-0.889569 - 5.99652 (0.723386 - τ) + 2.23449 EllipticPi [0.285116 ,
      JacobiAmplitude [2.70285 (0.723386 - τ), 0.782948 ], 0.782948 ]) +
 2.09657 (3.34259 (-0.0149885 + 0.125779 (0.723386 - τ) + 0.147072
        EllipticPi [1.1862 , JacobiAmplitude [2.70285 (0.723386 - τ), 0.782948 ], 0.782948 ]) +
    0.00908256 (0.0444135 + 0.165492 (0.723386 - τ) + 119.438 EllipticPi [556.454 ,
          JacobiAmplitude [2.70285 (0.723386 - τ), 0.782948 ], 0.782948 ])) -
 Abs[(78.0794 JacobiCN [2.70285 (-0.723386 + τ), 0.782948 ] × JacobiDN [
          2.70285 (-0.723386 + τ), 0.782948 ] × JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]) /
      (8.4482 - 2.40872 JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]^2) +
     (13.0208 JacobiCN [2.70285 (-0.723386 + τ), 0.782948 ] × JacobiDN [
          2.70285 (-0.723386 + τ), 0.782948 ] × JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]
        (0.362974 + 14.4439 JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]^2)) /
      (8.4482 - 2.40872 JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]^2)^2] /
   (5.99652 + (0.362974 + 14.4439 JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]^2) / (8.4482 - 2.40872 JacobiSN [2.70285 (-0.723386 + τ), 0.782948 ]^2))
```

In[298]:=
```
TemporalMotion [0.2 , 0, 10 , 0.3 , 0, -10 , 0, 0]
```
Out[298]=
```
tηm4Function [0.2 ,0,10 ,0.3 ,0,-10 ,0,0,<<>>]
```

In[299]:=
```
TemporalMotion [0.9 , 0, 2, 3, 0, -5, 10, 8]
```
Out[299]=
```
tηp3Function [0.9 ,0,2,3,0,-5,10,8,<<>>]
```

PLOTS OF THE GEODESICS

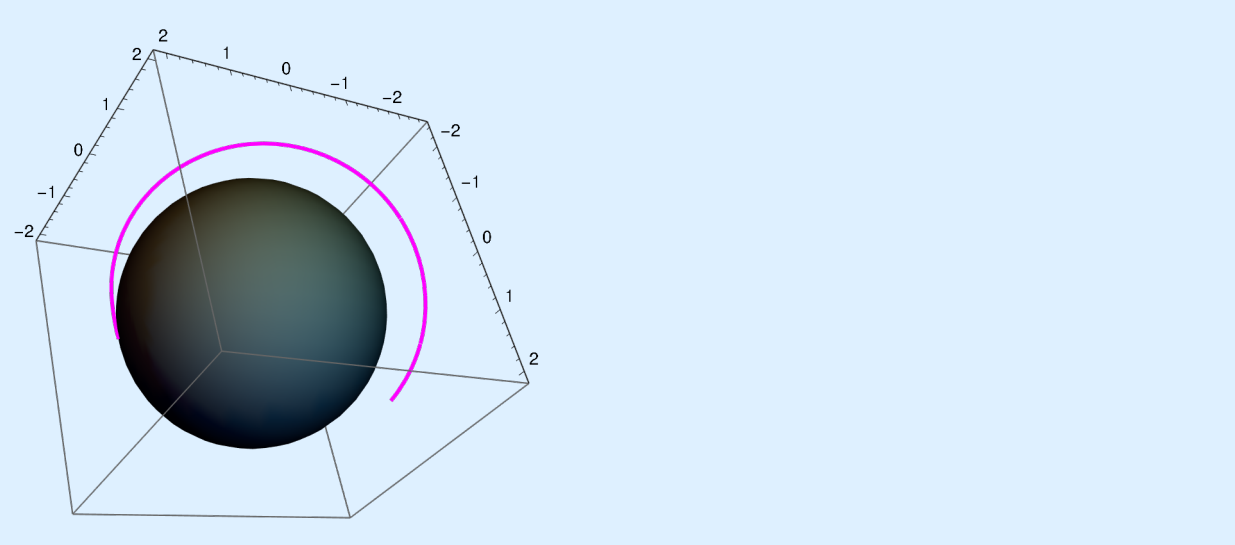
- NOTE: All Null geodesics that move into the black hole, we terminate them on the event horizon. The black sphere in the plots represents the event horizon.

Case one

In[301]:=
```
Module[
 {a = 0.3, ts = 0, rs = 2.5, θs = 2.6, ϕs = 0, prs = 2.7, pθs = 7.1, pϕs = 20, r, θ, ϕ, τ},
 θ = PolarMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion[a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D[
   r[τ]{Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta, PlotRange → All],
  Graphics3D[{Black, Specularity[.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
]
```

```
r,θ,ϕ={case1Function[0.3,0,2.5,2.6,0,2.7,7.1,20,<<>>],
  ordinaryFunction[0.3,0,2.5,2.6,0,2.7,7.1,20,<<>>],
  ϕηp1Function[0.3,0,2.5,2.6,0,2.7,7.1,20,<<>>]}
```

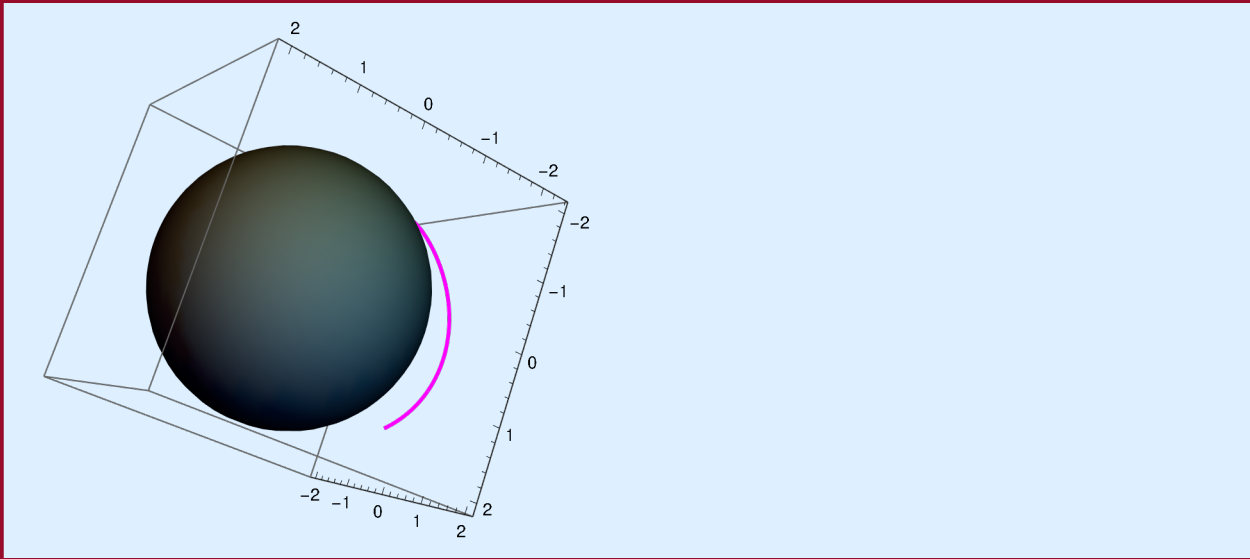
Out[301]=



In[302]:=
```
Module[{a = 0.3, ts = 0, rs = 2.5, θs = 2.6,
  ϕs = 0, prs = -2.7, pθs = 7.1, pϕs = 20, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ]{Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta, PlotRange → All],
  Graphics3D [{Black, Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
]
```

```
r,θ,ϕ={case1Function [0.3,0,2.5,2.6,0,-2.7,7.1,20,<<>>],
  ordinaryFunction [0.3,0,2.5,2.6,0,-2.7,7.1,20,<<>>],
  ϕηp1Function [0.3,0,2.5,2.6,0,-2.7,7.1,20,<<>>]}
```

Out[302]=

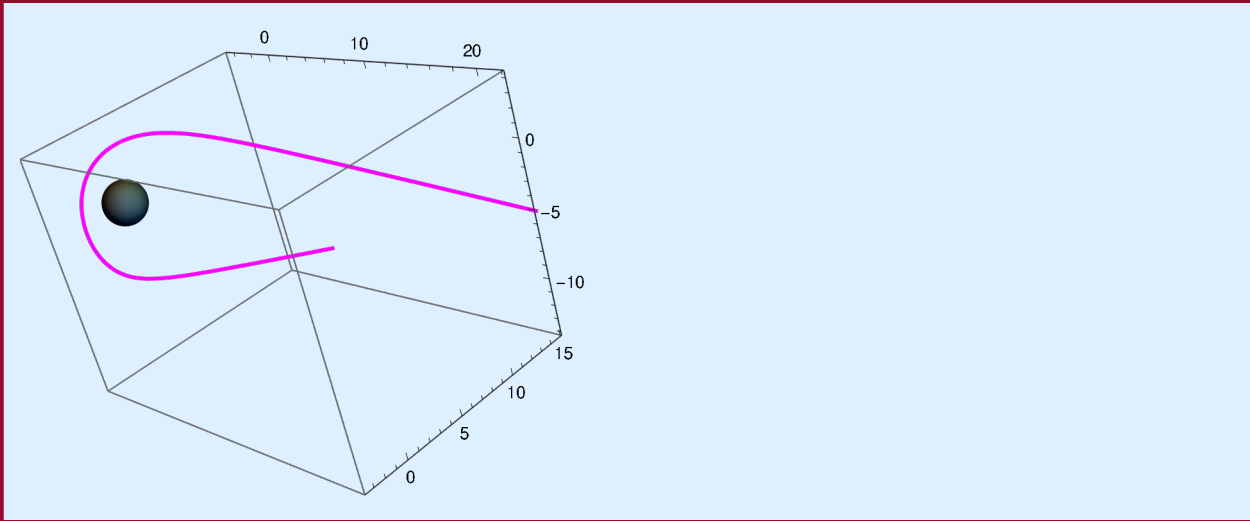


case two

In[303]:=
```
Module[{a = 0.8, ts = 0, rs = 21, θs = π / 2,
   ϕs = 0, prs = -1154.21, pθs = 13, pϕs = -10, r, θ, ϕ, τ},
  θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
     θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
  Show[
   ParametricPlot3D [
    r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
    {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta],
   Graphics3D [{Black, Specularity [.5], Sphere [{0, 0, 0}, 1 + √(1 - a^2)]}]
  ]
]
```

```
r,θ,ϕ={case2Function [0.8,0,21,Pi
--
2,0,-1154.21,13,-10,<<>>], ordinaryFunction [0.8,0,21,Pi
--
2,0,-1154.21,13,-10,<<>>], ϕηp2Function [0.8,0,21,Pi
--
2,0,-1154.21,13,-10,<<>>]}
```

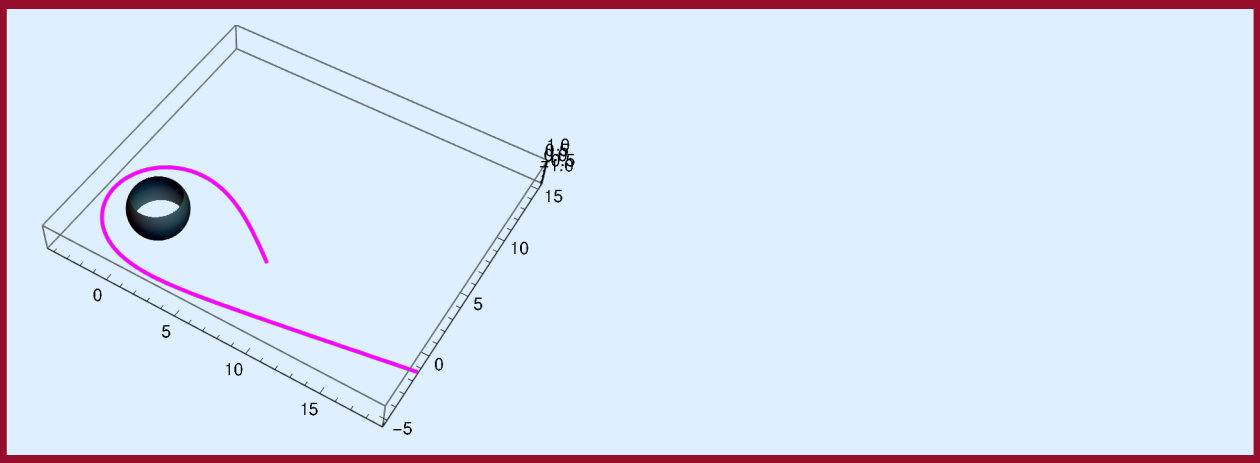
Out[303]=



In[305]:=
```
Module[{a = 0, ts = 0, rs = 8, θs = π / 2, ϕs = 0, prs = -2, pθs = 0, pϕs = 0.2 , r, θ, ϕ, τ},
  θ = PolarMotion [a, ts, rs, θs, ϕs, prs , pθs, pϕs];
  r = RadialMotion [a, ts, rs, θs, ϕs, prs , pθs, pϕs];
  ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs , pθs, pϕs];
  Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs , pθs, pϕs], PolarMotion [a, ts, rs,
      θs, ϕs, prs , pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs , pθs, pϕs]}];
  Show[
   ParametricPlot3D [
    r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
    {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs , pθs, pϕs]}, PlotStyle → Magenta ],
   Graphics3D [{Black , Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
  ]
]
```

```
r,θ,ϕ={case2Function [0,0,8,Pi
--
2,0,-2,0,0.2,<<>>], ordinarySchwarzchildFunction [0,0,8,Pi
--
2,0,-2,0,0.2,<<>>], ϕSchwarzchildFunction [0,0,8,Pi
--
2,0,-2,0,0.2,<<>>]}
```

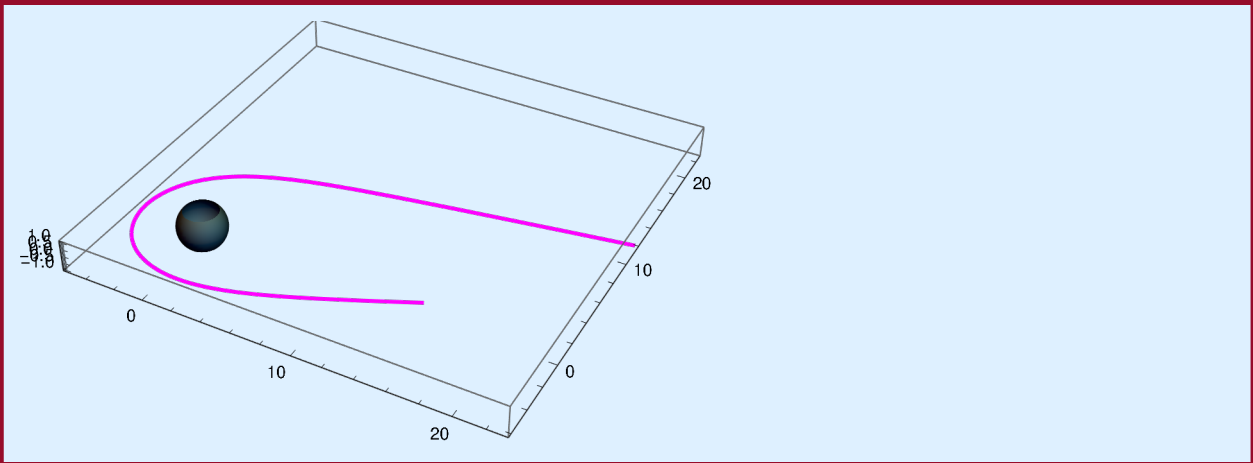
Out[305]=



In[306]:=
```
Module[{a = 0.8, ts = 0, rs = 15, θs = π / 2,
  ϕs = 0, prs = -312.74, pθs = 0, pϕs = -10, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta],
  Graphics3D [{Black, Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
]
```

```
r,θ,ϕ={case2Function [0.8,0,15,Pi
--
2,0,-312.74,0,-10,<<>>], ordinaryEquatorialFunction [0.8,0,15,Pi
--
2,0,-312.74,0,-10,<<>>], ϕ2EquatorialFunction [0.8,0,15,Pi
--
2,0,-312.74,0,-10,<<>>]}
```

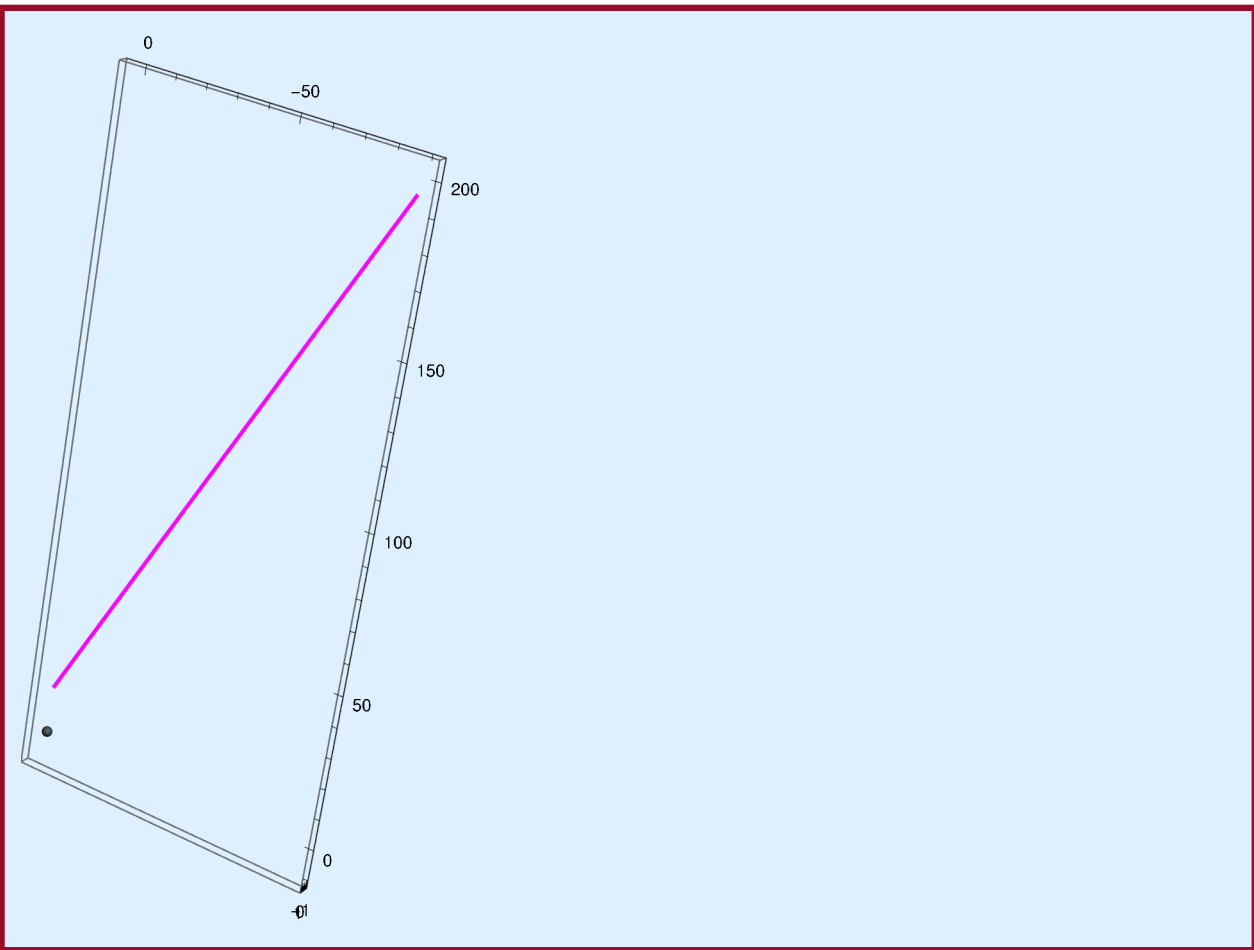
Out[306]=



```
Module [{a = 0.8 , ts = 0, rs = 15 , θs = π / 2,
   ϕs = 0, prs = 312.74 , pθs = 0, pϕs = -10 , r, θ, ϕ, τ},
  θ = PolarMotion [a, ts , rs , θs, ϕs , prs , pθs , pϕs];
  r = RadialMotion [a, ts , rs , θs, ϕs , prs , pθs , pϕs];
  ϕ = AzimuthalMotion [a, ts , rs , θs, ϕs , prs , pθs , pϕs];
  Print["r,θ,ϕ=", {RadialMotion [a, ts , rs , θs, ϕs , prs , pθs , pϕs], PolarMotion [a, ts , rs ,
     θs, ϕs , prs , pθs , pϕs], AzimuthalMotion [a, ts , rs , θs, ϕs , prs , pθs , pϕs]}];
  Show[
   ParametricPlot3D [
    r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
    {τ, 0, Maxτ[a, ts , rs , θs, ϕs , prs , pθs , pϕs]}, PlotStyle → Magenta , PlotRange → All],
   Graphics3D [{Black , Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
  ]
](*when prs>0, the geodesic goes to infinity without encoutering a turning point*)
```

```
r,θ,ϕ={case2Function [0.8,0,15,Pi
--
2,0,312.74,0,-10,<<>>], ordinaryEquatorialFunction [0.8,0,15,Pi
--
2,0,312.74,0,-10,<<>>], ϕ2EquatorialFunction [0.8,0,15,Pi
--
2,0,312.74,0,-10,<<>>]}
```

Out[307]=

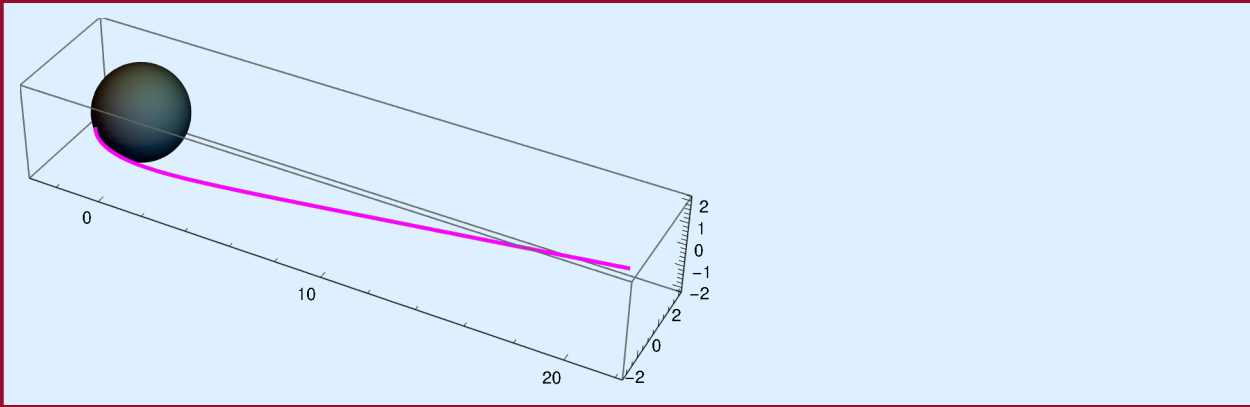


case three

In[308]:=
```
Module[{a = 0, ts = 0, rs = 21, θs = π / 2,
   ϕs = 0, prs = -1154.21 , pθs = 0, pϕs = -10, r, θ, ϕ, τ},
  θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
  Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
     θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
  Show[
   ParametricPlot3D [
    r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
    {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta , PlotRange → All],
   Graphics3D [{Black , Specularity [.5], Sphere [{0, 0, 0}, 1 + √(1 - a^2) ]}]
  ]
]
```

```
r,θ,ϕ={case3Function [0,0,21,Pi
--
2,0,-1154.21 ,0,-10,<<>>], ordinarySchwarzchildFunction [0,0,21,Pi
--
2,0,-1154.21 ,0,-10,<<>>], ϕSchwarzchildFunction [0,0,21,Pi
--
2,0,-1154.21 ,0,-10,<<>>]}
```

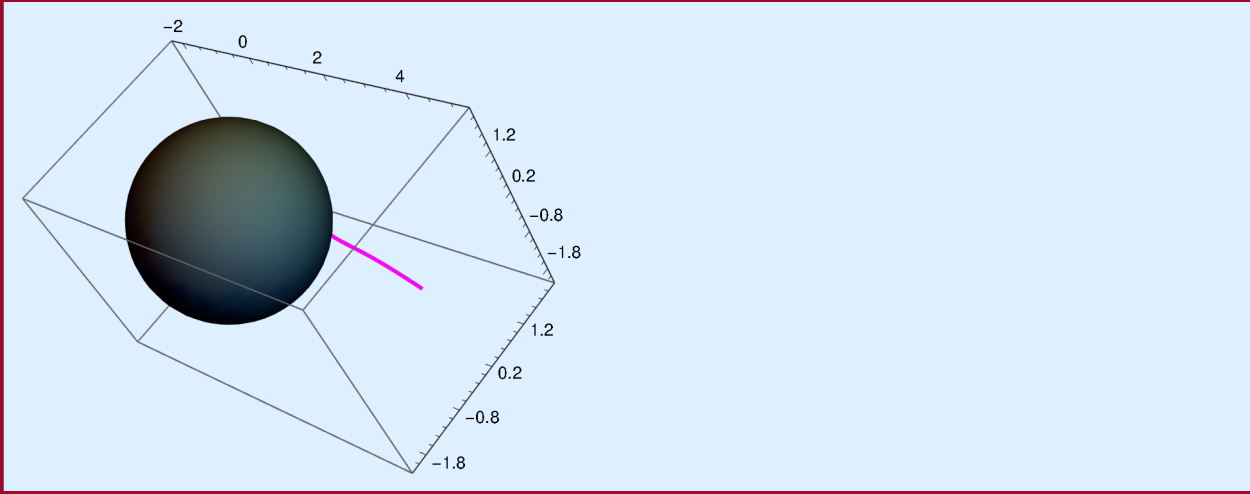
Out[308]=



In[309]:=
```
Module[{a = 0.2, ts = 0, rs = 5, θs = π/2, ϕs = 0, prs = -100, pθs = 10, pϕs = 8, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ]{Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta
   (*,PlotRange →{{-5,5},{-5,5},{-5,5}}*), PlotRange → All],
  Graphics3D [{Black, Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
]
```

```
r,θ,ϕ={case3Function [0.2,0,5,Pi
--
2,0,-100,10,8,<<>>], ordinaryFunction [0.2,0,5,Pi
--
2,0,-100,10,8,<<>>], ϕηp3Function [0.2,0,5,Pi
--
2,0,-100,10,8,<<>>]}
```

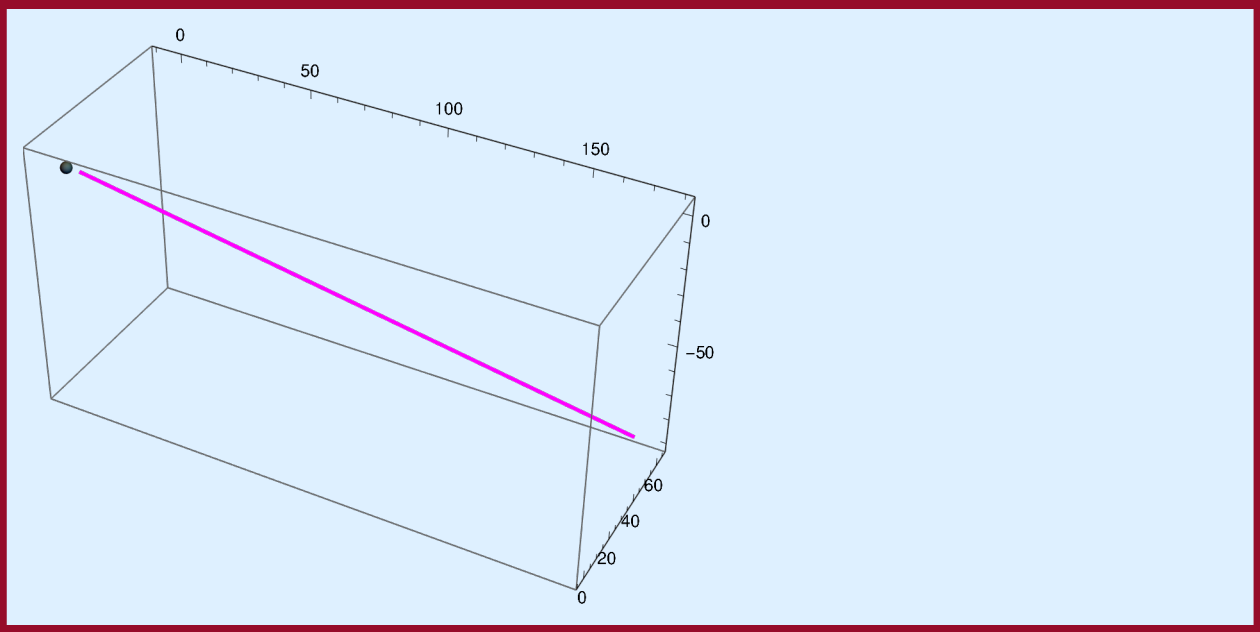
Out[309]=



```
Module[{a = 0.2, ts = 0, rs = 5, θs = π/2, ϕs = 0, prs = 100, pθs = 10, pϕs = 8, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ]{Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta
   (*,PlotRange →{{-5,5},{-5,5},{-5,5}}*), PlotRange → All],
  Graphics3D [{Black, Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
](*when prs>0, the geodesic goes to infinity without encoutering a turning point*)
```

```
r,θ,ϕ={case3Function [0.2,0,5,Pi
--
2,0,100,10,8,<<>>], ordinaryFunction [0.2,0,5,Pi
--
2,0,100,10,8,<<>>], ϕηp3Function [0.2,0,5,Pi
--
2,0,100,10,8,<<>>]}
```

Out[310]=

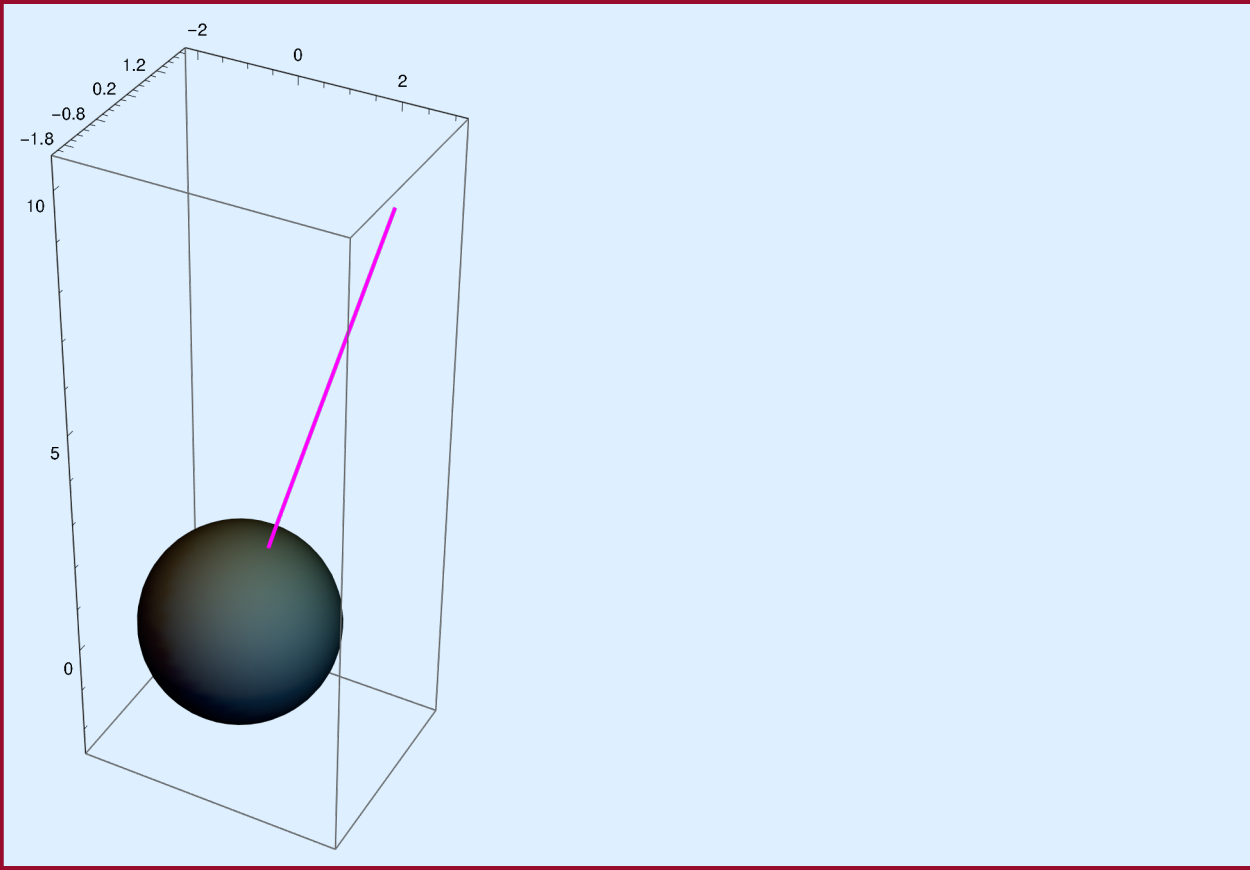


case four

In[312]:=
```
Module[{a = 0.2, ts = 0, rs = 10, θs = 0.3, ϕs = 0, prs = -10, pθs = 0, pϕs = 0, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[ϕ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta, PlotRange → All],
  Graphics3D [{Black, Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
]
```

```
r,θ,ϕ={case4Function [0.2,0,10,0.3,0,-10,0,0,<<>>],
  vorticalFunction [0.2,0,10,0.3,0,-10,0,0,<<>>], ϕηm4Function [0.2,0,10,0.3,0,-10,0,0,<<>>]}
```

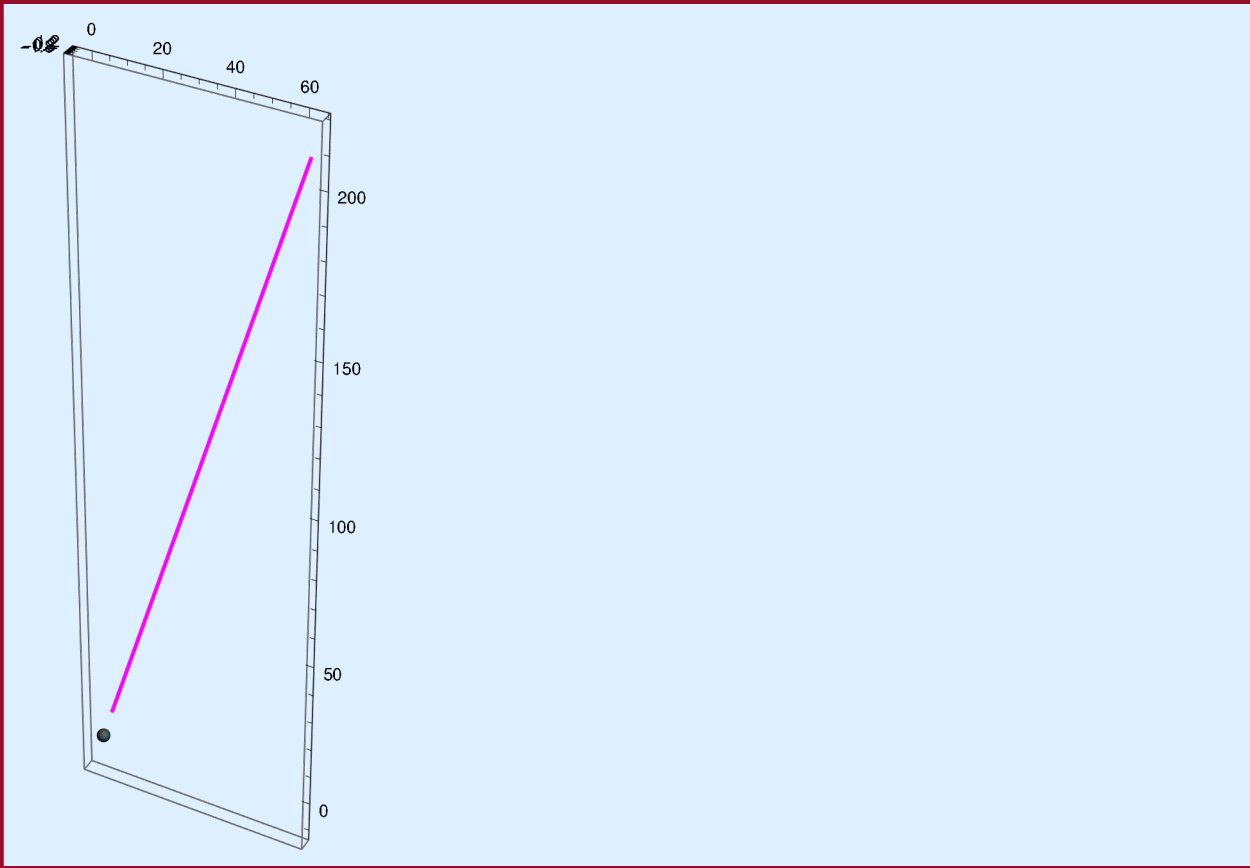
Out[312]=



```
Module[{a = 0.2, ts = 0, rs = 10, θs = 0.3, ϕs = 0, prs = 10, pθs = 0, pϕs = 0, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Print["r,θ,ϕ=", {RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs], PolarMotion [a, ts, rs,
    θs, ϕs, prs, pθs, pϕs], AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs]}];
 Show[
  ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[ϕ[τ]]},
   {τ, 0, Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta, PlotRange → All],
  Graphics3D [{Black, Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}]
 ]
](*when prs>0, the geodesic goes to infinity without encoutering a turning point*)
```

```
r,θ,ϕ={case4Function [0.2,0,10,0.3,0,10,0,0,<<>>],
  vorticalFunction [0.2,0,10,0.3,0,10,0,0,<<>>], ϕηm4Function [0.2,0,10,0.3,0,10,0,0,<<>>]}
```

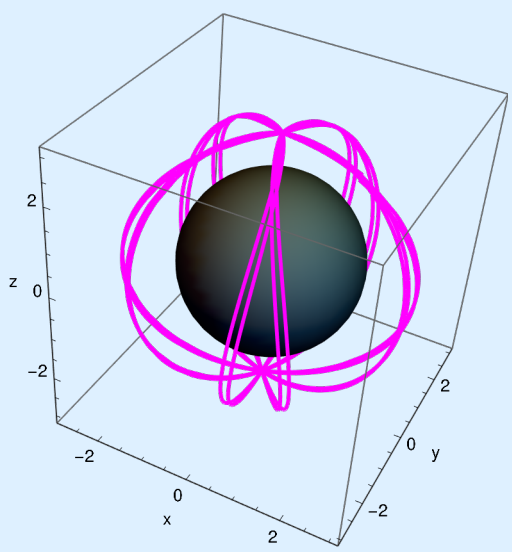
Out[313]=



Spherical geodesics

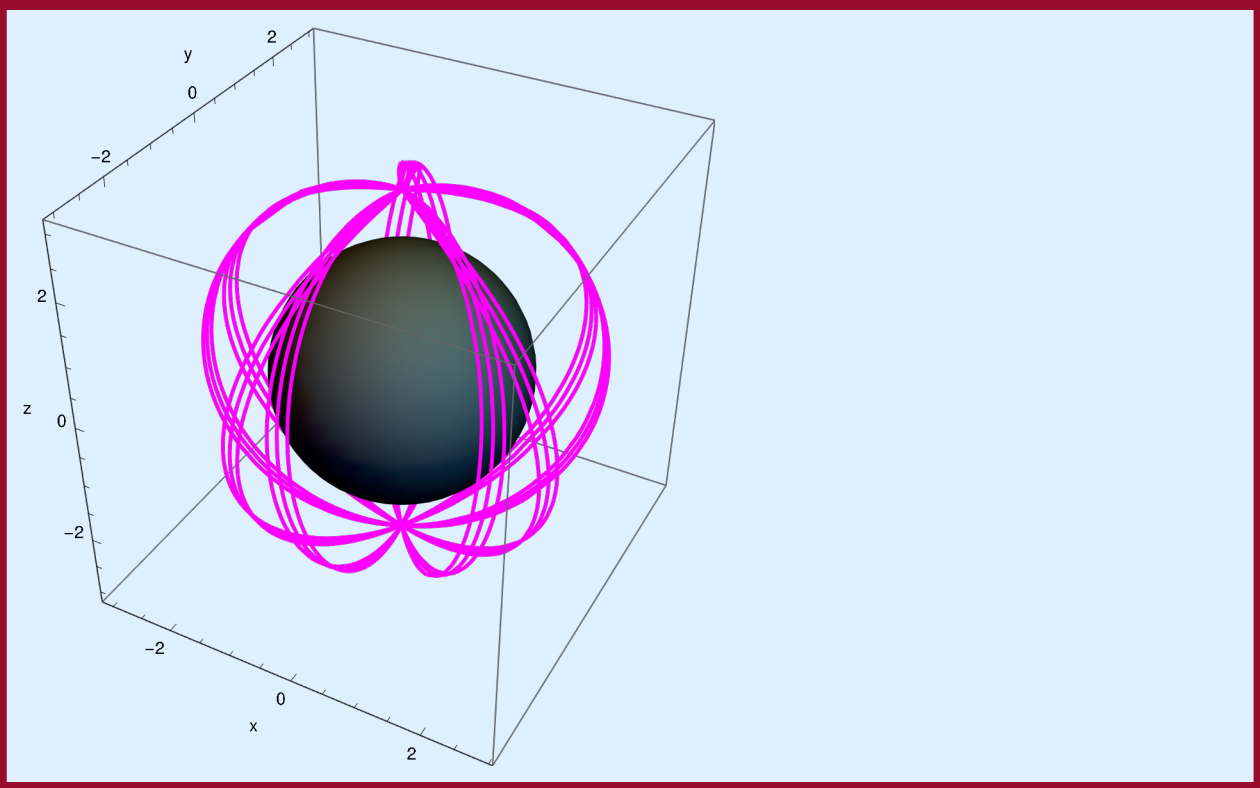
In[315]:=
```
Module[{a = 0.5`32 , ts = 0, rs = 2.8832177419263525`32 ,
  θs = 0, ϕs = 0, prs = 0, pθs = 20, pϕs = 10, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 KerrNullGeoRadialRoots [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Show[ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, 10 Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta , PlotRange → All],
  Graphics3D [{Black , Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}] ,
  AxesLabel → {"x", "y", "z"}]
]
```

Out[315]=



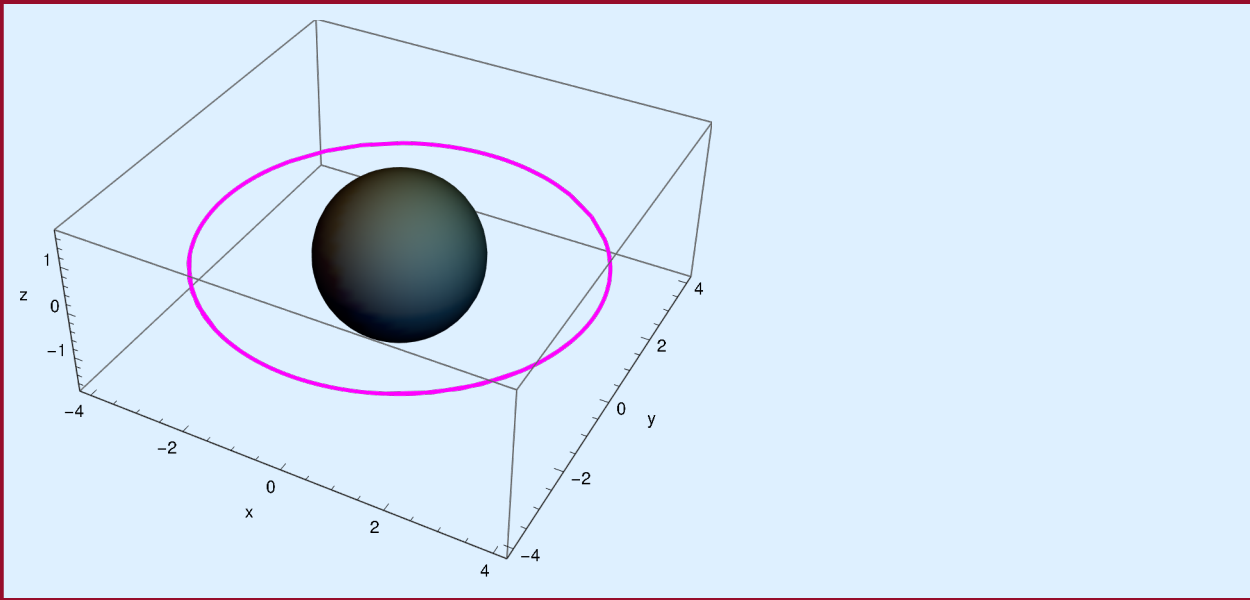
In[316]:=
```
Module[{a = 0.5`32 , ts = 0, rs = 2.8832177419263525`32 ,
  θs = π, ϕs = 0, prs = 0, pθs = 20, pϕs = 10, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Show[ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, 15 * Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]},
   PlotStyle → Magenta , PlotRange → All], Graphics3D [
   {Black , Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}], AxesLabel → {"x", "y", "z"}]
]
```

Out[316]=



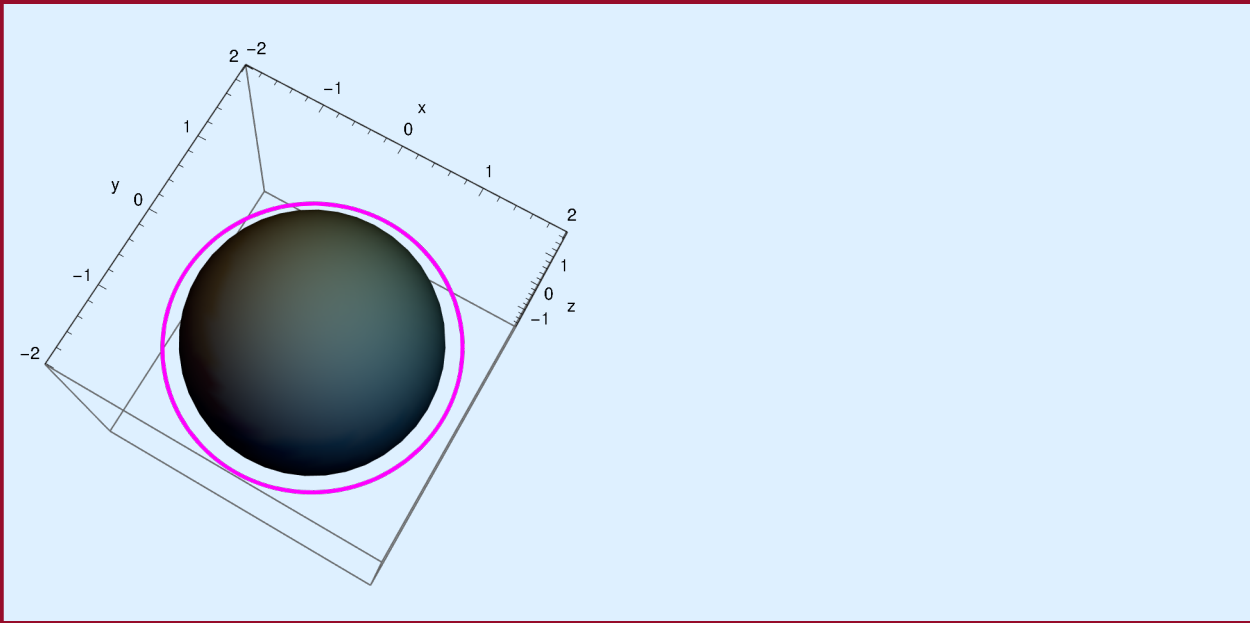
In[317]:=
```
Module[{a = 0.8, ts = 0, rs = 2 (1 + Cos[2/3 ArcCos[0.8]]),
  θs = π/2, ϕs = 0.5, prs = 0, pθs = 0, pϕs = -10, r, θ, ϕ, τ},
 θ = PolarMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion[a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Show[ParametricPlot3D[
   r[τ]{Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, 2 Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta, PlotRange → All],
  Graphics3D[{Black, Specularity[.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}],
  AxesLabel → {"x", "y", "z"}]
]
```

Out[317]=



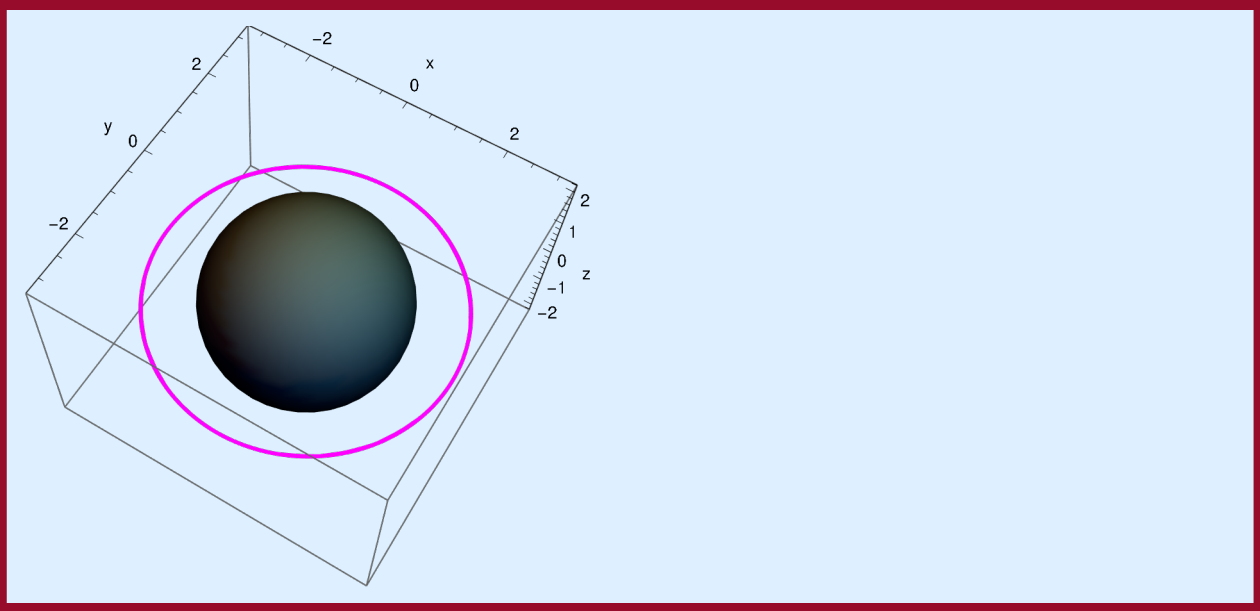
```
Module[{a = 0.8`32 , ts = 0, rs = 2 (1 + Cos[4 π/3 + 2/3 ArcCos[0.8`32 ]]),
  θs = π / 2, ϕs = 0.5`32 , prs = 0, pθs = 0, pϕs = 10, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Show[ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, 2 Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta , PlotRange → All],
  Graphics3D [{Black , Specularity [.5], Sphere [{0, 0, 0}, 1 + Sqrt[1 - a^2]]}] ,
  AxesLabel → {"x", "y", "z"}]
]
```

Out[318]=



```
Module[{a = 0, ts = 0, rs = 3, θs = π / 2, ϕs = 0.5`32 , prs = 0, pθs = 0, pϕs = 10, r, θ, ϕ, τ},
 θ = PolarMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 r = RadialMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 ϕ = AzimuthalMotion [a, ts, rs, θs, ϕs, prs, pθs, pϕs];
 Show[ParametricPlot3D [
   r[τ] {Cos[ϕ[τ]] × Sin[θ[τ]], Sin[ϕ[τ]] × Sin[θ[τ]], Cos[θ[τ]]},
   {τ, 0, 2 Maxτ[a, ts, rs, θs, ϕs, prs, pθs, pϕs]}, PlotStyle → Magenta , PlotRange → All],
  Graphics3D [{Black , Specularity [.5], Sphere[{0, 0, 0}, 1 + Sqrt[1 - a^2]]}] ,
  AxesLabel → {"x", "y", "z"}]
](*Schwarzschild  photon  sphere *)
```

Out[319]=



- These have been examples of the null geodesics the users can study and they can input different values of their choice to study more properties.

CONCLUSION

- We have formulated the code such that the user only needs a given set of initial position and momentum.
- Given the initial position and momenta, the user will be able to study various properties of spherical and non spherical null geodesics in Kerr and Schwarzschild space-time.
- We hope that this work together with other works in the literature will enable us to understand the nature of black holes in more detailed ways.
- **NOTE: We have not yet made the code publicly available as we are still concluding of various tests**

THANK YOU!
OBRIGADA!
SHUKRANI!
MBUYA MONO!